

⑩ 日本国特許庁 (JP)

⑪ 特許出願公開

⑫ 公開特許公報 (A)

昭55—137360

⑬ Int. Cl.³

識別記号

庁内整理番号

⑭ 公開 昭和55年(1980)10月27日

F 02 P 5/04

8011—3G

F 02 D 5/02

6933—3G

35/00

7604—3G

// B. 60 R 16/02

6839—3D

発明の数 1
審査請求 未請求

(全 6 頁)

⑮ 自動車用制御装置

横浜市神奈川区西寺尾町714

⑯ 特 願 昭54—45324

⑰ 出 願 人 日産自動車株式会社

⑱ 出 願 昭54(1979)4月16日

横浜市神奈川区宝町2番地

⑲ 発 明 者 安保敏巳

⑳ 代 理 人 弁理士 中村純之助

明 細 書

1. 発明の名称 自動車用制御装置

2. 特許請求の範囲

1. ストアド・プログラム方式のデジタル計算機を用いて車載機器を制御する自動車用制御装置において、基本的構成が同一で一部分のみが異なる複数のプログラムを一つのサブルーチンとして構成し、上記サブルーチンの実行に先立って該サブルーチンの使い方を指定することにより、一部分が異なる複数のプログラムの演算を一つのサブルーチンで共通に行なわせるように構成した自動車用制御装置。

2. 上記サブルーチンとして、内燃機関の運転条件に応じて制御量を予めセットされたデータテーブルからルックアップするプログラムすなわち、テーブル・ルックアップ・ルーチンを用い、データテーブルの格納番地及びそのデータ量を指定してからテーブル・ルックアップ・ルーチンへジャンプさせることにより、データ量の異なる複数の

のデータテーブルのルックアップを一つのテーブル・ルックアップ・ルーチンで共通に行なわせることを特徴とする特許請求の範囲第1項記載の自動車用制御装置。

3. 上記サブルーチンとして、入力データの演算処理を行なうプログラムを用い、入力データの有効桁数や小数点位置を示す形式、処理後のデータの形式及び演算処理に使用する定数のうちの少なくとも一つを指定してジャンプさせることにより、形式等の異なるデータの演算を一つのサブルーチンで共通に行なわせることを特徴とする特許請求の範囲第1項記載の自動車用制御装置。

4. 上記サブルーチンとして、データに同一処理を複数回繰返して行なうプログラムを用い、繰返し回数を指定してジャンプさせることにより、繰返し回数の異なるデータ処理を一つのサブルーチンで共通に行なわせることを特徴とする特許請求の範囲第1項記載の自動車用制御装置。

5. ジャンプする行先番地を指定することによって繰返し回数を選択することを特徴とする特許

請求の範囲第4項記載の自動車用制御装置。

3. 発明の詳細な説明

本発明はストアド・プログラム方式のデジタル計算機（いわゆるマイクロコンピュータ）を用いて各種の車載機器を制御する自動車用制御装置に関し、特にプログラムのサブルーチンに関するものである。

最近、マイクロコンピュータを用いて自動車の各種車載機器、例えば内燃機関（燃料供給量、点火時期、排気還流量、回転速度等の制御）、変速機、ブレーキ、ラジオ、メータ類等を制御する装置が開発されている。

上記のごとき制御装置のプログラミングの手法として、同一の処理が何度も繰返して行なわれるものについては、サブルーチンが良く用いられる。

例えば、第1図の様なフローチャートの場合、処理A→処理J→処理B→処理J→処理Cの順番で実行されるが、これを第2図のように処理Jをサブルーチンとし、処理Aを実行後、J S R（サブルーチンにジャンプせよ）という命令を挿入す

れば処理Jを実行し、R T S（サブルーチンからリターンせよ）という命令でもとに戻り、続いて処理Bを実行する。再びJ S Rにより処理Jを実行し、R T Sで処理Cを実行する。これにより処理Jを2回プログラムする必要はなくなり、プログラムの量を削減することが出来るので、メモリを節約することが出来る。

しかし、上記のごとき従来のサブルーチンにおいては、全く同一のプログラムはサブルーチン化出来るが、プログラムの構成は良く似ていても、その一部（扱うデータのビット数、扱うデータの量、繰返し処理の回数等）が異なる場合には、サブルーチン化することが出来なかった。例えば、第3図のJとJ'の構成が非常に似たものであっても、データのビット数等が異なるだけでサブルーチン化することが出来ないという問題があった。

本発明は上記の問題に鑑みてなされたものであり、サブルーチンにジャンプする前にそのサブルーチンの使い方を指定してからジャンプさせるように構成することにより、一部が異なるプログラ

・ 3 ・

ムをもサブルーチン化出来るようにし、プログラムの量を更に削減できるようにした自動車制御装置を提供することを目的とする。

以下図面に基づいて本発明を詳細に説明する。

第4図は本発明を適用する自動車用制御装置の一例のブロック図である。

第4図において、1は被制御機器であり、例えば内燃機関である。また2は制御装置であり、入出力装置3、中央演算装置（CPU）4、読出し専用メモリ（ROM）5、読出し書き込みメモリ（RAM）6等から構成されている。

内燃機関1からは、機関運転状態を示す各種の信号（回転速度信号、吸入空気量信号、水温信号、スロットル弁全閉スイッチ等の各種スイッチの信号等）が入出力装置3を介してCPU4に送られる。CPU4はROM5に記憶されているプログラムに従った演算を行ない、各種の制御信号（燃料供給量、点火進角、排気還流量等を制御する信号）を入出力装置3を介して内燃機関1に送って、所望の制御を行なう。

・ 5 ・

次に第5図は本発明の一実施例図であり、(イ)はフローチャート、(ロ)はアキュムレータを示す。

第5図の実施例は、8ビットのマイクロコンピュータ（例えばモトローラ6800）において、CPU内のアキュムレータA、B（図ではAcc.A及びAcc.Bと表示、各々8ビット）の内容を二つ組合せて16ビットのデータとして用いる場合に、アキュムレータA、Bの内容のシフトの回数を選択できるサブルーチンを示す。

すなわち、アキュムレータの内容を右へシフトするルーチンR₁、R₂、R₃を縦に接続し、ジャンプする位置を指定することによって、シフトの回数を選択出来るようにしたものであり、例えばSHIFT 3へジャンプすれば、サブルーチンが終了（R T S命令まで）するまでにシフトを3回行なう。同様にSHIFT 2、SHIFT 1へジャンプすれば、それぞれ2回、1回シフトされる。

したがって第5図のサブルーチンを用い、ジャンプする位置を指定してやれば、一つのサブルーチンで異なるシフト回数の演算を自由に選択し

・ 6 ・

て行なわせることが出来るので、シフト回数の異なる複数のプログラムを備えておく必要がなくなる。

次に、第6図は本発明の第2の実施例のフローチャートである。

第6図の実施例は、例えばA/D変換された入力データに演算処理を行なうサブルーチン(サブルーチン名はCAL)で、定数÷A/D変換データという割算を実行して、演算結果を8ビット精度で出す場合と10ビット精度で出す場合とを一つの割算ルーチンで行なうものを示す。

第6図(イ)において、8ビット精度データをA/D変換すると、FLAG(FLAGという名前を付けたRAM(メモリ))を0にしてCALへジャンプする。また、第6図(ロ)において10ビット精度データをA/D変換するとFLAGを1にしてCALへジャンプする。次に第6図(ハ)において、サブルーチンCALではその実行に先立って入力データが10ビット精度か8ビット精度かをFLAGによって判断し、10ビット精度(

FLAG=1)の場合、被除数である定数を2ビット左へシフトして8ビット精度の場合と条件を同じにして割算を実行すれば、同一の割算ルーチンで両方を実行することが出来る。

上記のようなデータの形式(有効桁数や小数点位置)や、また更に演算処理に使用する定数が異なる場合でもサブルーチン化することが出来る。

次に、第7図は本発明の第3の実施例のフローチャートである。

この実施例は、エンジンの運転条件(回転数、¹⁰ 負荷量)に応じて2次元テーブルから最適制御量(進角、燃料補正量)をルックアップするサブルーチン(テーブル・ルックアップ・ルーチン)である。まず実施例の説明に先立ってテーブル・ルックアップ・ルーチンについて、第8図に基づいて説明する。第8図(イ)は、エンジン回転数と負荷量から点火進角をルックアップするデータテーブルの1例図、(ロ)は、これをROMにストアした例を示す図である。(イ)のデータテーブルの横軸 $N_1 \sim N_8$ は例えばエンジン回転数であり縦軸 $L_1 \sim L_8$

$\sim L_8$ は負荷量である。今エンジン回転数 N が $N_1 < N < N_8$ で負荷量 L が $L_1 < L < L_8$ であったとすると、それぞれに対応するデータ $D_{11}, D_{12}, D_{13}, D_{14}$ を用いて補間計算をすることにより回転数 N 、負荷量 L に対応する進角値 D が計算出来る。

実際の手順は、計測された負荷量 L をもとにして $DEDE$ 番地(E は16進法の14、 D は16進法を示す)から順に格納してある $L_1 \sim L_8$ を使って $L_1 < L < L_8$ となることを判別し、続いて、回転数 N をもとにして $DEDE$ 番地から順に格納してある $N_1 \sim N_8$ を使って $N_1 < N < N_8$ となることを判別し、この二つの結果から、 $DEDE$ から格納してあるメインテーブルの $D_{11}, D_{12}, D_{13}, D_{14}$ を使って補間計算を行なう。本例は、回転数、負荷量を、共に8分割した 8×8 のテーブルについて行なったが、例えば16分割の 16×16 のテーブルについても同様である。

次に第7図(ロ)において、点火時期を演算するルーチン(ADV)は、回転数(8分割)、負荷量(8分割)に応じてデータテーブルから進角値

をルックアップする。この場合、 8×8 テーブルのルックアップを示すため、SIZE(SIZEという名前をつけたRAM(メモリ))を0にしておく。また点火進角の場合、トランスミッションの形式に応じて点火進角を変えるため、データテーブルの格納番地が異なっている。このためMT(マニュアルトランスミッション)かAT(オートマチックトランスミッション)かに応じて、データテーブルの格納番地を T_1, T_2, T_3 (それぞれ名前を付けたRAM)にセットしてLOOKUPルーチンへジャンプするようにしておく。また第7図(ハ)において、燃料補正量を演算するルーチンは、回転数、負荷量共16分割で精度を上げて 16×16 のテーブルルックアップを行なう。これを示すためSIZEを1にしてさらにデータテーブルの格納番地を $T_1 T_2 T_3$ にセットしてLOOKUPルーチンへジャンプする。

T_1, T_2, T_3 のセットは、具体的には第7図(ニ)に示す様に、メモリ(ROM)内に縦軸格子点データ S_i (例えば前記 $L_1 \sim L_8$)、横軸格子

点データS。(例えば前記N₁~N₈)、メインテーブル8。(8×8あるいは16×16)が格納されているとすると、T1にB E B B B、T2にB E B B B、T3にB E B B B(但しBは1.6進法を示す)をセットすることになる。

次に第7図(イ)のLOOK UP ルーチンでは、S I Z Eをチェックして8×8のテーブル・ルックアップか16×16のテーブル・ルックアップかを判断しそれに応じてテーブル・ルックアップを行なう。

上記のように第7図の実施例においては、8×8のテーブル・ルックアップと、精度の高い16×16のテーブル・ルックアップとを一つのサブ・ルーチンで行なうことが出来る。

以上説明したごとく本発明においては、サブ・ルーチンの使い方を指定してからジャンプさせることにより、構成の似たルーチンは全てまとめて一つのサブ・ルーチンにすることが出来るので、プログラムの量、したがってメモリの量を大巾に節約することが出来るという効果がある。

4. 図面の簡単な説明

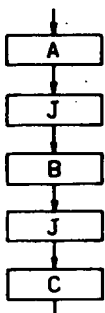
第1図~第3図は従来例のフローチャート、第4図は本発明を適用する自動車用制御装置の一例図、第5図~第7図はそれぞれ本発明の実施例のフローチャート、第8図はデータテーブルの一例図である。

符号の説明

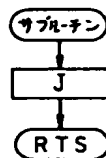
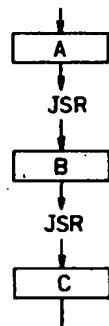
- | | |
|-------------|------------|
| 1 ... 被制御機器 | 2 ... 制御装置 |
| 3 ... 入出力装置 | 4 ... CPU |
| 5 ... ROM | 6 ... RAM |

代理人弁理士 中 村 純之助

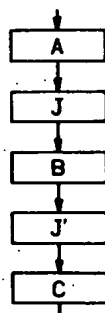
オ 1 図



オ 2 図



オ 3 図



オ 4 図

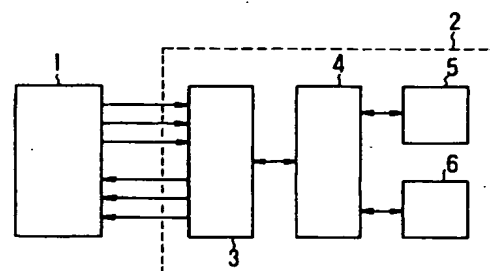


表 8 図

(4)

L ₀	D ₀₀	D ₀₁	D ₀₂	D ₀₃	D ₀₄	D ₀₅	D ₀₆	D ₀₇	D ₀₈
L ₁	D ₁₀	D ₁₁	D ₁₂	D ₁₃	D ₁₄	D ₁₅	D ₁₆	D ₁₇	D ₁₈
L ₂	D ₂₀	D ₂₁	D ₂₂	D ₂₃	D ₂₄	D ₂₅	D ₂₆	D ₂₇	D ₂₈
L ₃	D ₃₀	D ₃₁	D ₃₂	D ₃₃	D ₃₄	D ₃₅	D ₃₆	D ₃₇	D ₃₈
L ₄	D ₄₀	D ₄₁	D ₄₂	D ₄₃	D ₄₄	D ₄₅	D ₄₆	D ₄₇	D ₄₈
L ₅	D ₅₀	D ₅₁	D ₅₂	D ₅₃	D ₅₄	D ₅₅	D ₅₆	D ₅₇	D ₅₈
L ₆	D ₆₀	D ₆₁	D ₆₂	D ₆₃	D ₆₄	D ₆₅	D ₆₆	D ₆₇	D ₆₈
L ₇	D ₇₀	D ₇₁	D ₇₂	D ₇₃	D ₇₄	D ₇₅	D ₇₆	D ₇₇	D ₇₈
L ₈	D ₈₀	D ₈₁	D ₈₂	D ₈₃	D ₈₄	D ₈₅	D ₈₆	D ₈₇	D ₈₈
L ₉	D ₉₀	D ₉₁	D ₉₂	D ₉₃	D ₉₄	D ₉₅	D ₉₆	D ₉₇	D ₉₈

(□)

(7F1X)

SE000

SE020

SE040

(b)

SE050

SE060

SE070

L ₁	L ₂	L ₃	L ₄	L ₅	L ₆	L ₇	L ₈
N ₁	N ₂	N ₃	N ₄	N ₅	N ₆	N ₇	N ₈
D ₀₀	D ₀₁	D ₀₂	D ₀₃	D ₀₄	D ₀₅	D ₀₆	D ₀₇
D ₁₀	D ₁₁	D ₁₂	D ₁₃	D ₁₄	D ₁₅	D ₁₆	D ₁₇
D ₂₀	D ₂₁	D ₂₂	D ₂₃	D ₂₄	D ₂₅	D ₂₆	D ₂₇
D ₃₀	D ₃₁	D ₃₂	D ₃₃	D ₃₄	D ₃₅	D ₃₆	D ₃₇
D ₄₀	D ₄₁	D ₄₂	D ₄₃	D ₄₄	D ₄₅	D ₄₆	D ₄₇
D ₅₀	D ₅₁	D ₅₂	D ₅₃	D ₅₄	D ₅₅	D ₅₆	D ₅₇
D ₆₀	D ₆₁	D ₆₂	D ₆₃	D ₆₄	D ₆₅	D ₆₆	D ₆₇
D ₇₀	D ₇₁	D ₇₂	D ₇₃	D ₇₄	D ₇₅	D ₇₆	D ₇₇
D ₈₀	D ₈₁	D ₈₂	D ₈₃	D ₈₄	D ₈₅	D ₈₆	D ₈₇